Gateway Design for Computer Network Interconnection

M. Beeler, J. Burchfiel, R. Rettberg,
V. Strazisar, and D. Walden

Bolt Beranek and Newman Inc.
Cambridge, Massachusetts USA

PRTN #
156

October 1975

Contents

Preface 3

1. Introduction 4

2. Gateways as Hosts vs. Gateways as Nodes 7

3. Gateway Characteristics 11

4. The Gateway Virtual Network 18

5. Division of Functions Between the Gateway and TCP 22

6. A Prototypical Implementation 26

7. Debugging and Coordination 35

8. Backwards Compatibility 38

Appendix A -- Relationship Between Host Routing and Internetwork Routing 40

Appendix B -- Fragmentation 44

Appendix C -- An example of network specific functions 47

References 51

Figure 1 -- Networks Connected by Gateways 5
Figure 2 -- An Internetwork Segment Embedded in a Network Message 10
Figure 3 -- Gateway Virtual Network 19
Figure 4 -- Components of a Gateway 27
Figure 5 -- Organization of ARPANET-PRN Gateway 33
Figure 6a -- PRN Host 34
Figure 6b -- ARPANET Host 34

Preface

What follows should be regarded as a working document rather than a complete and final report. Feedback we have received from readers suggests that this document could be improved by specifying gateway implementation in greater detail. We feel, however, that the report is at a development plateau and publication as it stands is preferable to further delay. We expect to expand the report, or issue a separate companion report, to present detailed gateway specifications.

Digitized by the Internet Archive
in 2025 with funding from
Amateur Radio Digital Communications, Grant 151

https://archive.org/details/temporary-note-156

1. Introduction

The work done to date on the interconnection of computer networks has usually assumed a configuration such as that shown in figure 1. (In this report, our discussion refers only to networks of the packet-switching variety, and assumes the reader to be familiar with packet-switching terminology.) On each network there are hosts (denoted by H in the figure) which desire to communicate with hosts on other networks. The networks are connected together by units (denoted by G) called "gateways."* The gateways must in some way convert traffic in the format of one network into traffic in the format of another network.

Because host-to-host protocols differ from one network to the next, and because these protocols are generally complicated and incompatible, many researchers (including ourselves) believe that hosts on different networks wishing to communicate must do so in a common protocol. Much of the work to date in network interconnection has been to specify such a standard protocol: see for example [Cerf 74a], [Zimmerman 74], [McKenzie 74b]. While there has not yet been agreement on the standard protocol, for the purposes of this report we assume the terminology and

---

*Note that this is a different use of the term "gateway" than in conventional international telephone system interconnection, where the term is used to refer to an artificial site from which it is convenient to establish tariffs).

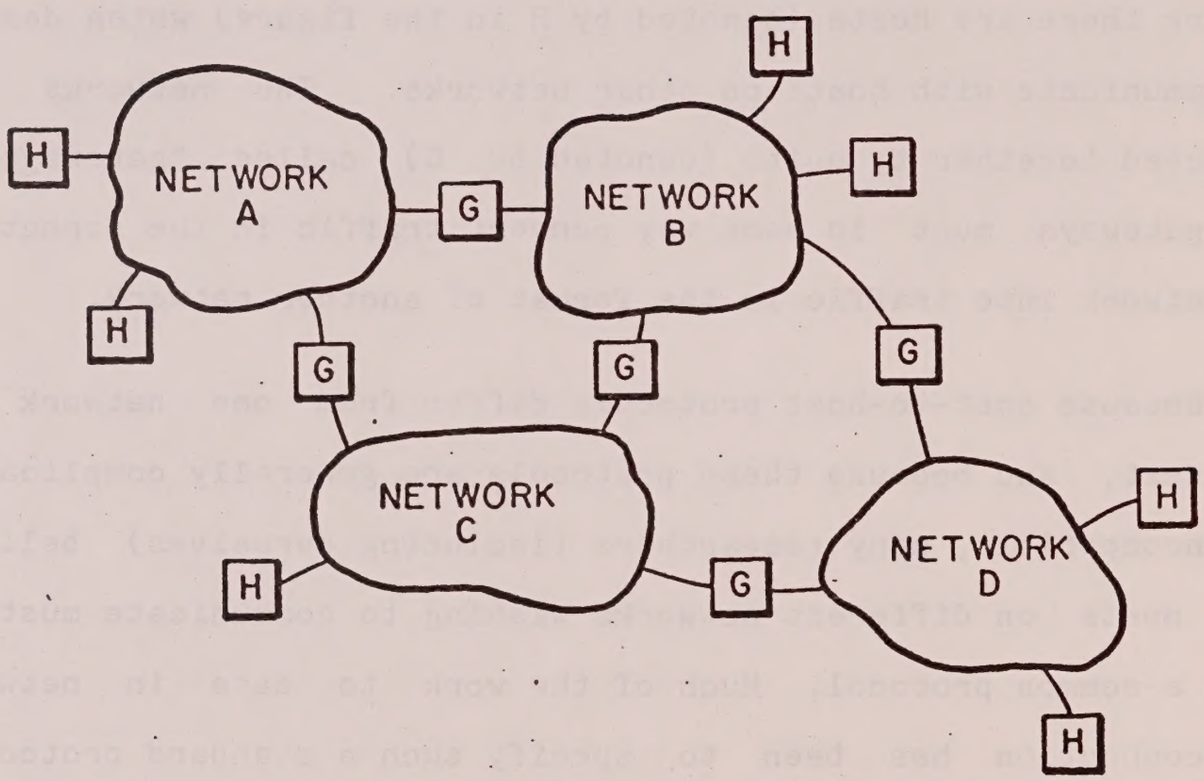


Figure 1 -- Networks Connected by Gateways

protocol described by Cerf and Kahn [Cerf 74a], [Cerf 74b]. In this protocol the logical entity in the host which performs the protocol functions is called the Transmission Control Program or TCP.

While there has been considerable work on the standard host protocol, there has been less work on the function and structure of the gateway. For the most part it has been assumed that the gateway will forward traffic between networks (and across networks) without specifying how this would be done. In this report we consider the functions the gateway should perform and how it should perform them.*

---

*[Burchfiel 74], [Lloyd 75], and [Belloni 74] have also addressed this issue.

2. Gateways as Hosts vs. Gateways as Nodes

One of the outstanding questions of network interconnection is whether the gateways should connect networks at the packet or host level. By packet level, we mean that a portion of the gateway would actually become a node on each of the networks being connected. By host level, we mean that a portion of the gateway would actually be a host on each of the networks being connected. We feel that the gateways should connect at the host level primarily to maintain the sovereignty of the networks involved [Crowther 72]. Furthermore, it is unlikely that a standard packet format can be found, at the present state of ongoing development of all of the packet-switching networks that might be connected, which would permit packet level connection.

By network sovereignty, we mean that connection to the networks must be done at a point where the interface is both well defined and well controlled by the constituent networks. If the point of connection is the host level just mentioned, each network can protect itself against activities of the gateway to the same extent as it may protect itself against the activities of any other host.

As already mentioned, we believe that it will be impossible, in general, for the gateway to convert between the host-to-host

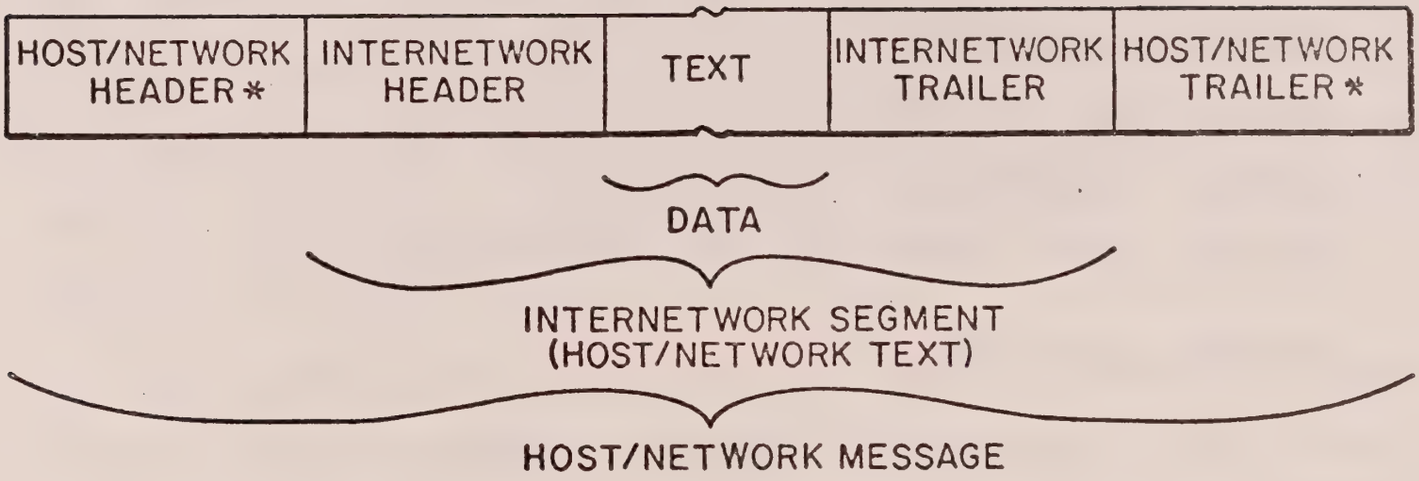
protocols of two communicating networks. Thus, rather than communicating in the host-to-host protocol of the network, the gateways should communicate with nodes of the network in the lowest form of host/network protocol supported by the network. Transmissions in the network interconnection protocol of the TCPs should be the text of these host/network protocol messages.

Notice that a host on a given network might find itself having to implement, in addition to the host-to-host protocol of its own network, the standard internetworking host-to-host protocol for communication with hosts in other networks. Of course, one can hope the internetwork standard will eventually prevail throughout the world and the host-to-host protocols of the individual networks will eventually wither away.

To summarize, hosts involved in internetwork communications must adopt a common protocol, and gateways should connect to networks as hosts using the lowest level of host/network protocol. Further, protocols have already been specified [e.g., Cerf 74a] for the former task and the protocol for the latter task is specified by the network for any network to which a gateway is to be attached [e.g., BBN 74].

If the gateways connect to the networks as hosts, then the format of the messages passed to the network is specified by the

host/network protocol. This protocol is then used to permit transparent transmission of segments of an internetwork transmission. This can be done by embedding the internetwork segment in the text of a message in the host/network protocol as shown in figure 2. Such a composite message has two headers and potentially two trailers. The outermost header and trailer provide information for the network. The header will specify the address of the gateway host to which the message should be delivered, any allocation or sequencing information which is used by the host/network protocol, and any further information demanded by that protocol. An example of a trailer that might be required by the host/network protocol would be padding and a checksum. Within this outermost header and trailer is the internetwork data segment with its header and trailer. The internetwork header specifies such information as the ultimate destination, sequencing, and reassembly information. The actual data which is being transferred is the text of this message.



*THIS HEADER VARIES FROM NETWORK TO NETWORK

Figure 2 -- An Internetwork Segment Embedded in a Network Message

3. Gateway Characteristics

We will now examine some of the characteristics a gateway must have in addition to being able to pass messages between two networks. The gateway must have:

- -a capability for inter-gateway routing
- -access control mechanisms
- -accounting mechanisms
- -statistics gathering
- -a capability for fragmentation
- -control of congestion at the gateways
- -in some cases, a capability for inter-gateway retransmission
- -checks and balances

In the following paragraphs we elaborate on each of these points.

Routing. Inter-gateway routing is desirable for all of the standard reasons one desires routing. For example, for reliability one must have alternate paths over which traffic may be routed; for achieving higher bandwidth than is available over any single path, one wants to be able to route traffic over parallel paths; different classes of traffic should be able to follow different routes (e.g., traffic requiring low delay should be routed around networks which insert large delays).

Access Control. A given constituent network may wish to limit some classes of traffic or all traffic at some times (e.g., because of regulatory considerations, country A might not want traffic from country B to country C to pass through country A's network). Also, a given user might not wish his traffic to pass through some constituent networks. For example, the U.S. Defense Department might allow its traffic to go to England via Canada but not via Cuba for obvious political reasons. These limitations may be provided by declaring traffic to belong to one or another routing class. In any case, the constituent networks are very likely to want to monitor the use of their network by internetwork traffic.

For efficiency of routing and access control, in large networks with hundreds of hosts and gateways, the routing algorithm will probably need to have a hierarchical structure knowing about logical and/or physical areas.

Logical subsets of the entire collection of hosts can be useful not only for routing but also to provide a virtual subnet feature. Groups of hosts wishing only to communicate among themselves and desiring isolation from irrelevant or malicious traffic from other hosts may declare themselves to be a logical entity called a virtual subnet. Gateways could recognize this distinction and prevent internetwork traffic from entering or

leaving the group of hosts. The virtual subnet might be essentially static, as in divisions of a company, or highly dynamic, as in a telephone conference call.

Accounting. In the simplest case accounting is merely counting the number of packets received from each network and those transmitted to each network connected to the gateway. Billing may accrue for use of a network as a gateway-to-gateway packet transmission mechanism, in which case the network specific portion of the gateway appropriately keeps the records. Billing for use of the central gateway functions is also conceivable, in which case the central functions would record usage. More complex accounting is possible. Charges may vary with class of service, size of packet, prevailing traffic level, and so forth. Further study of accounting from a business standpoint is needed. This is one area where variation from network to network may require routines specific to the network or to the combination of networks the gateway connects.

Accounting information may be accessible to the user. In particular, gateways can supply information on the rates currently in effect. The user then has the option of choosing a different class of service or deferring his use of the facilities until rates become more favorable.

Statistics gathering. Optimal use of the network is facilitated by adequate statistics gathering. The gateways collect data on their own performance and that of the networks they interconnect. These statistics may be periodically forwarded to a coordination center, where further performance measures can be extracted. These collected statistics are then available both to gateway maintenance personnel, who may adjust parameters to optimize gateway performance, and to users, who may select the manner of network use which will best suit their needs.

Fragmentation. Because of the differences in maximum message size of the constituent networks connected by a gateway, the gateway must have the ability to fragment a larger message arriving from one network into smaller messages which are acceptable by the next network. When such fragmentation occurs, the message stream must eventually be reassembled into its original structure. The protocol proposed in [Cerf 74a] provides the reassembly function at the destination host.

Congestion Control. Congestion will inevitably occur at the gateway unless specific measures are taken to prevent it. This congestion can occur as a result of speed mismatches between the networks connected by a gateway, because several gateways on a network may simultaneously transmit traffic to the same other

gateway, because traffic may have to be held during a period of recovery from a failure, and so on. One specific kind of congestion results from deadlocks, such as when gateway A is full of traffic for gateway B which is full of traffic for gateway A.

Retransmission. When a message is lost in the network between two gateways, one can either retransmit the message between the two gateways or assume that the message will be retransmitted from the source host to the destination host. It has been shown [NAC 73] that hop-to-hop retransmission is more efficient than source-to-destination retransmission if the possibility of message loss is appreciable; and even when there is little possibility of message loss, the variance of retransmission delays is less with hop-to-hop retransmission than with source-to-destination retransmission. While some networks deliver messages very reliably, other networks (e.g., the packet radio network) rely on source-to-destination retransmission and in some cases are quite cavalier about throwing away messages. Thus, the gateways should have the ability to retransmit messages across lossy networks. It seems that at least when the hosts on a network are normally responsible for retransmission across that network, the gateways ought to provide retransmission across that network. ([Mader 74] supports the notion that end-to-end timeout and retransmission can be unduly inefficient.)

Some consideration has been given to adding a facility to gateways for filtering duplicate packets generated by the network spanned by the gateways. The filtering of such duplicates has obvious advantages in cases where large numbers of duplicate packets are generated in that such duplicates, if passed on to other networks, represent an undesirable load. However, there are also problems with trying to filter duplicates between gateways. First, it is difficult to catch the duplicates when some may go through one gateway and others through another. Second, it is difficult to distinguish undesired duplicates from required retransmissions. Further study of this issue is needed.

Checks and Balances It has been brought to our attention that constituent networks may require safeguards against traffic from other networks for not only technical but also political reasons. A violation of traffic class restrictions, or swamping by excessive traffic, are examples of such issues. Further, the gateway is obviously the proper place to include such safeguards to minimize both trouble and finger pointing. We propose that the gateway monitor and where possible prevent such abuses through a system of checks and balances. However, at this stage of gateway design we do not know what specific form these checks and balances should take. The best we can do at present is to keep the issue of checks and balances always in mind and

implement specific measures as seems appropriate. We welcome comment and suggestions from others on the areas and means of checks and balances implementation.

4. The Gateway Virtual Network

Notice that the characteristics of a gateway described above are very similar to the characteristics of a node on a packet-switching network [Crowther 75]. This leads one to the notion of a gateway virtual network wherein the gateways act as nodes and the network spanned by the gateways acts as virtual lines fully connecting all the gateways on that network. This concept is illustrated in figure 3, where network C has been replaced by the virtual lines it provides. Further, notice in the figure that logically there is a gateway associated with each host attempting internetwork communication.

Although the figure shows a separate gateway for each host, we do not mean to imply that the gateways must be physically separate machines or that there need necessarily be a one-to-one correspondence between hosts and gateways. For instance, the logical entity that is the gateway may take the form of a program running in a host computer. Alternately, the gateway could be in a stand-alone machine connecting two networks or serving one or more hosts. On the other hand the gateway connecting two networks could even take the form of a program running in a host which is connected to both networks. In general, a gateway should be able to connect any combination of hosts and networks. (The political and economic issues relating to whether the

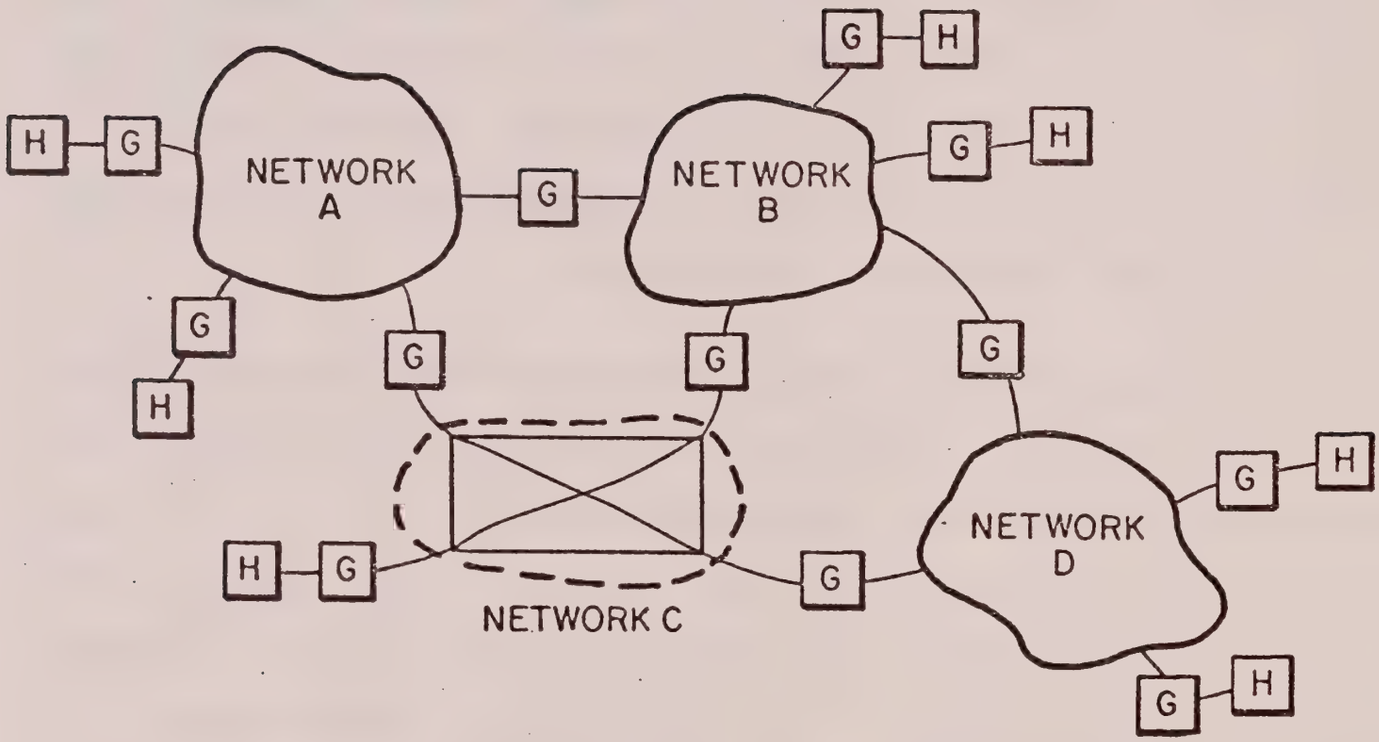


Figure 3 -- Gateway Virtual Network

gateway should reside in a host or a separate gateway machine are discussed in [Kuo 75].) Furthermore, it is entirely possible for a host not to have its own gateway at all, preferring to use a gateway elsewhere in its network to perform the gateway functions for the host. In this case, the host would simply know the address of a couple of gateways in its network and would send its internetwork traffic arbitrarily to one of these gateways for routing and forwarding. Note that the gateway arbitrarily chosen may not be on the best path to the destination, and this gateway may, in fact, forward the traffic to another (better) gateway in the same network for forwarding outside the network. Although this approach may be inefficient, it reduces the number of gateways that have to be constructed. Further, we believe both approaches should be supported in the long term; since the two approaches are compatible, this causes no problems.

It is our view that the gateway virtual network should have many of the maintenance characteristics of a stand-alone packet-switching network [McKenzie 72]. It should have centralized development and maintenance responsibility, including a gateway network monitoring and coordination center (GCC). We do not, however, feel that the gateway program must run in any one brand of machine; indeed, it will probably be necessary to support the gateway program on machines of several nationalities because of the international extent of these networks.

The gateway virtual network is a general solution to the problem of interconnecting networks which is not highly dependent on the nature of the networks being connected. Because of this we expect that networks which are connected in the future will not require modification to facilitate connection. While one can conceive that a simpler but less general form of interconnection may be possible through modification of the current networks, future networks may be much less amenable to such an attachment and therefore require major modifications to themselves or even to the previously interconnected networks. Furthermore, the gateway virtual network, with its adaptive routing, congestion control, and gateway-to-gateway retransmission capability provides for high performance host-to-host communication across multiple networks (note that this approach is in contrast to [Opderbeck 74], which advocates sacrificing such efficiency for simplicity).

5. Division of Functions Between the Gateway and TCP

Just as in a stand-alone network there is a question of the division of responsibility between the hosts and the nodes (i.e., should reassembly be done by the hosts or should the nodes deliver traffic in order), in the gateway virtual network there is a question of the division of responsibility between the TCP and the gateway. It is best to implement these functions in either the gateway or the TCP, rather than blurring the implementation across the boundary between them. Some points are quite clear. For instance, reassembly must be done by the destination TCP, as traffic traversing the virtual network may be fragmented and pieces routed on alternate paths to the destination. Thus, the ultimate destination TCP must be prepared to rearrange the communication stream into the correct order and reassemble it into the internetwork data segment. (The reassembly problem is further aggravated when intervening networks do not maintain ordering.)

Second, the differences in message or packet size between constituent networks mean that it may be necessary for messages to be successively fragmented into smaller and smaller units as they pass from one network to the next. It seems natural that this task should be done in the gateway since it knows the message and packet size characteristics of the networks to which

it is connected. Even at the source host, the TCP can leave any necessary fragmentation of the message stream to the gateway.

Just as clearly, the gateways must be responsible for the routing calculation, since only the gateways have the global knowledge necessary to make a sensible routing decision.

The access control and accounting functions should also be in the gateway since these functions are desired between the networks.

It is less clear whether the gateways or TCPs should perform the congestion control and retransmission functions. One alternative would be for the gateways not to worry about retransmission and to solve any congestion problems simply by discarding traffic, in each case relying on the source and destination TCPs to provide the necessary recovery mechanisms ([Belsnes] advocates this). In fact, the TCP of [Cerf 74a] does provide such recovery mechanisms. On the other hand, if much traffic is discarded by the gateways to control congestion, or if there is even one network which loses traffic frequently, then we believe relying on source-to-destination TCP retransmission will be prohibitively inefficient and will also be expensive, both in direct (network-imposed accounting) and indirect (TCP overhead) costs. For this reason we think it is incumbent on the gateways

to shoulder the burden of controlling their own congestion and for the gateways to provide the option of retransmission across a lossy network. Of course, even though the gateways provide these functions, the TCPs should retain the end-to-end retransmission capability at their level for reliability, since retransmissions performed by the gateway level are for efficiency rather than for complete reliability.

The remainder of the functions performed by the [Cerf 74a] TCP are properly the functions of the TCP and not the gateways, as these functions are concerned with end-to-end issues, user process level issues, etc., while the gateways are properly concerned only with traffic switching issues.

There does have to be some communication between the TCP level and the gateway level. Most obviously, the TCP must specify the address of the destination TCP to which the gateways are to route the traffic. This particular communication can be effected simply by having the gateways understand the TCP traffic formats which include such addressing information. Another area of communication required between the TCPs and gateways is to specify certain transmission characteristics for the traffic (e.g., networks or geographical areas through which the traffic must not be routed, maximum acceptable delays, and average throughput required over a period of time). Communication

between TCPs can also specify traffic classes of priority service or logical groups of hosts (virtual subnets). This area has not been explicitly addressed previously and requires further study.

At present we have no opinion on whether the security function should go in the gateway or the TCP. Current U.S. military communications security standards appear to require pairwise encryption between source host and destination host. If such standards are maintained, then the security function must obviously reside with the TCPs. Alternatively, there has been much discussion of so-called "link encryption." Use of link encryption would lead one to place the security function in the gateways. A third alternative would be to place the security function between the source and destination TCPs and their gateways. In this case the TCP "header" must be in the clear, permitting the gateways to access the information. There are elaborations of all the above schemes and other schemes are possible.

6. A Prototypical Implementation

A prototypical implementation of the gateway is illustrated below. It consists of modular structures which carry on communications at a message level with various networks, modules which transform a particular network message format into a TCP type packet and vice versa, and modules which perform message routing and other centralized message processing functions such as flow and access control.

The purpose of each of these elements of the gateway will be clear if we follow a message as it is processed by the gateway from Network 1 to Network 2 in figure 4. The line to Network 1 is operated by the code labelled "N1 Network Specific Functions." This code acts as a host on that network, passing the required control information and transferring messages to and from other hosts on that network according to the lowest level message protocols of that network. In the case of a connection to the ARPA Network, this code would implement the standard Host/IMP protocol [BBN 74]. Incoming messages received from Network 1 are passed to the "Network 1 Stripper," where the Network 1 message header is removed. The result is an internetwork protocol segment. This segment is passed to the central portion of the gateway, which examines the address specified in the segment and queues it for output to the appropriate network. The "Network 2

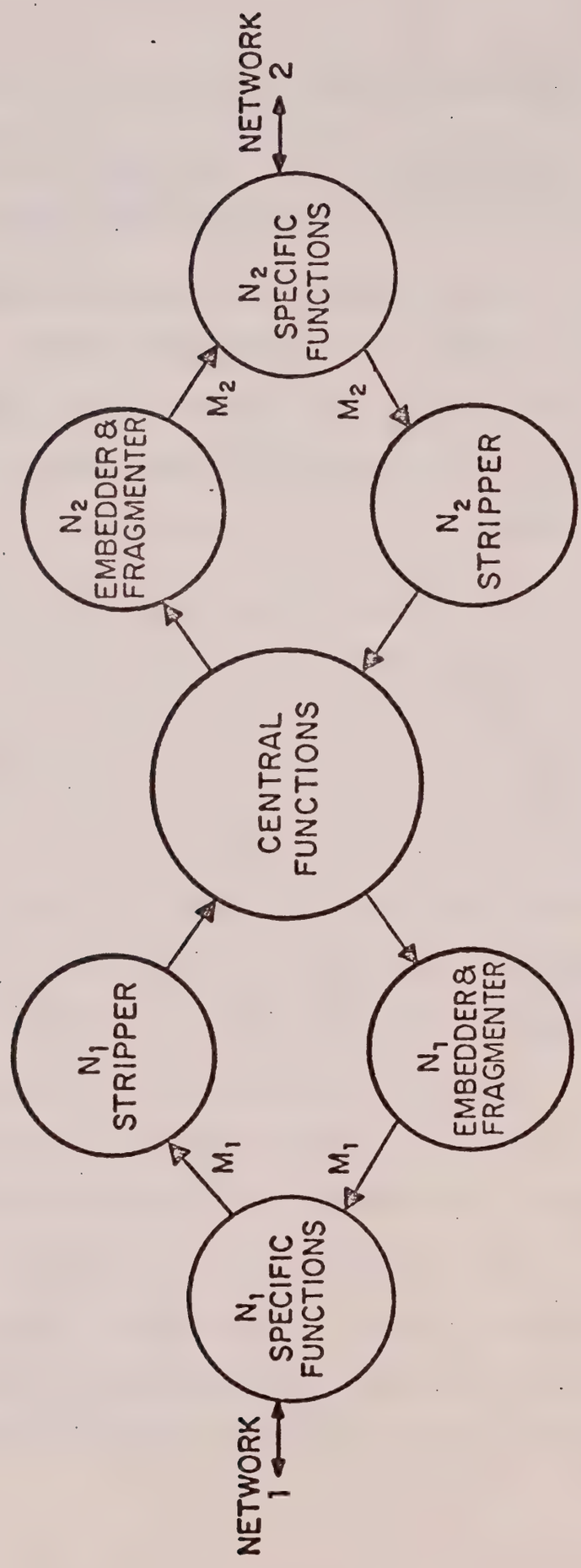


Figure 4 -- Components of a Gateway

Embedder and Fragmenter" takes these internetwork segments and converts them into messages in the Network 2 format. In this case the format will again be ARPA Network messages. If the internetwork segment does not fit within the text of a network message, it will be fragmented and re-formatted until all parts fit into network messages. Finally, the message is presented to the Network 2 message interface for delivery to the appropriate gateway host.

The control function will be responsible both for determining where to route an individual internetwork segment and for performing the distributed routing functions in cooperation with the other gateway nodes. In operation, one possible way to accomplish this distributed routing is as follows. Each gateway exchanges routing packets, containing information about all gateways in all networks, with every other gateway on each network to which it is attached. The gateway is given the host addresses of all other gateways on each of the networks to which it is attached in an internal table which will be used eventually for the access control logic. This is similar to the way an ARPA Network IMP exchanges routing packets with its immediate neighbors, where the packets contain information about every possible destination [McQuillan 74]; the other gateways on the same network are logically the immediate neighbors in the gateway

network. In this case, however, it is necessary for each gateway to contain internally the network address of the other gateways in the same network, because while an ARPA Network IMP has at most five neighbors, a network has potentially many hundreds of hosts, a large fraction of which may be gateways. Routing algorithms which require less storage in the gateways are also possible.

The information passed in the routing messages can contain at least four types of measurement about the network: delay, delay variability, bandwidth, and packet loss rate. For instance, gateways can accumulate delay and estimate bandwidth for the network connection between each pair of gateways in the same network. The gateways can then maintain a structure similar to that used by the ARPA IMP for routing. For each potential destination gateway, the intermediate gateway to which messages should be addressed for minimum delay and maximum bandwidth can be determined along with the expected delay and bandwidth. Periodically, the gateway reports to each of the other gateways on the same network the expected delay and bandwidth via this gateway. The other gateway then adds the delay to that gateway and determines the minimum of the bandwidth through that gateway and the bandwidth on the path to that gateway, to determine the delay and bandwidth which may be expected along this path to the

destination. Through this mechanism, each gateway can have the necessary information to route packets. Priority letters could be routed for delay, while others could be routed for bandwidth.

We recognize that this sample routing scheme suffers from an inability to accurately determine the expected delay and bandwidth along a path, particularly as the actual path across a network may change unbeknownst to the gateways. This is intrinsic to the implementation of gateways as hosts rather than as nodes, since only the nodes of a network normally have the necessary state information (e.g., line speeds between node pairs or the number of packets queued for a line), and so far, no network passes this information to its hosts. We believe that the reasons mentioned previously in this note for connecting gateways at the host level are very important, and that the way to improve the routing efficiently between gateways is for the networks to pass expected delay and bandwidth information, for example, to their hosts. Even without the issue of gateway routing, it might be useful for hosts on a network to receive such delay and bandwidth information to improve their use of the network (e.g., deciding whether to send a large file now or later).

We have suggested that in general some gateway related but network specific network control functions (such as local

addressing, connection establishment) will be required in association with each TCP. Since these functions, together with the actual gateway functions, form a network of their own, routing and formatting packets, and since there may eventually be many implementations of them, it is advisable to keep them as identical as possible. We therefore suggest that these routines be specified in some reasonably universally available higher level language. If possible, the network specific function code should be machine compiled from that specification. For instance, BCPL [BCPL 74] would be a reasonable choice of a high level language since much of the existent gateway and TCP code has already been written in BCPL. Otherwise, the routines can be hand compiled from the higher level language specification. We hope that in this way, the implementation effort and variability will be minimized in implementations at the various hosts.

Implementation of the gateway may be phased rather than concentrated into one effort. For instance, in initial tests where packets are merely transferred from one network back to the same network, the gateway may merely translate message headers, e.g., from a destination format to a source format. The first phase of full scale implementation will include:

-internet/local network interface(s)

-routing (first fixed and then dynamic)

-sufficient debugging capability

-rudimentary statistics gathering facility

The second phase will consist of:

-fragmentation

Components of the third phase of implementation are:

-access control

-accounting functions

-routing expanded to classes of traffic

-expanded statistics and performance monitoring

Figures 5, 6a and 6b illustrate the functional modules of a gateway as applied to a specific gateway between the ARPA network and the Packet Radio network.

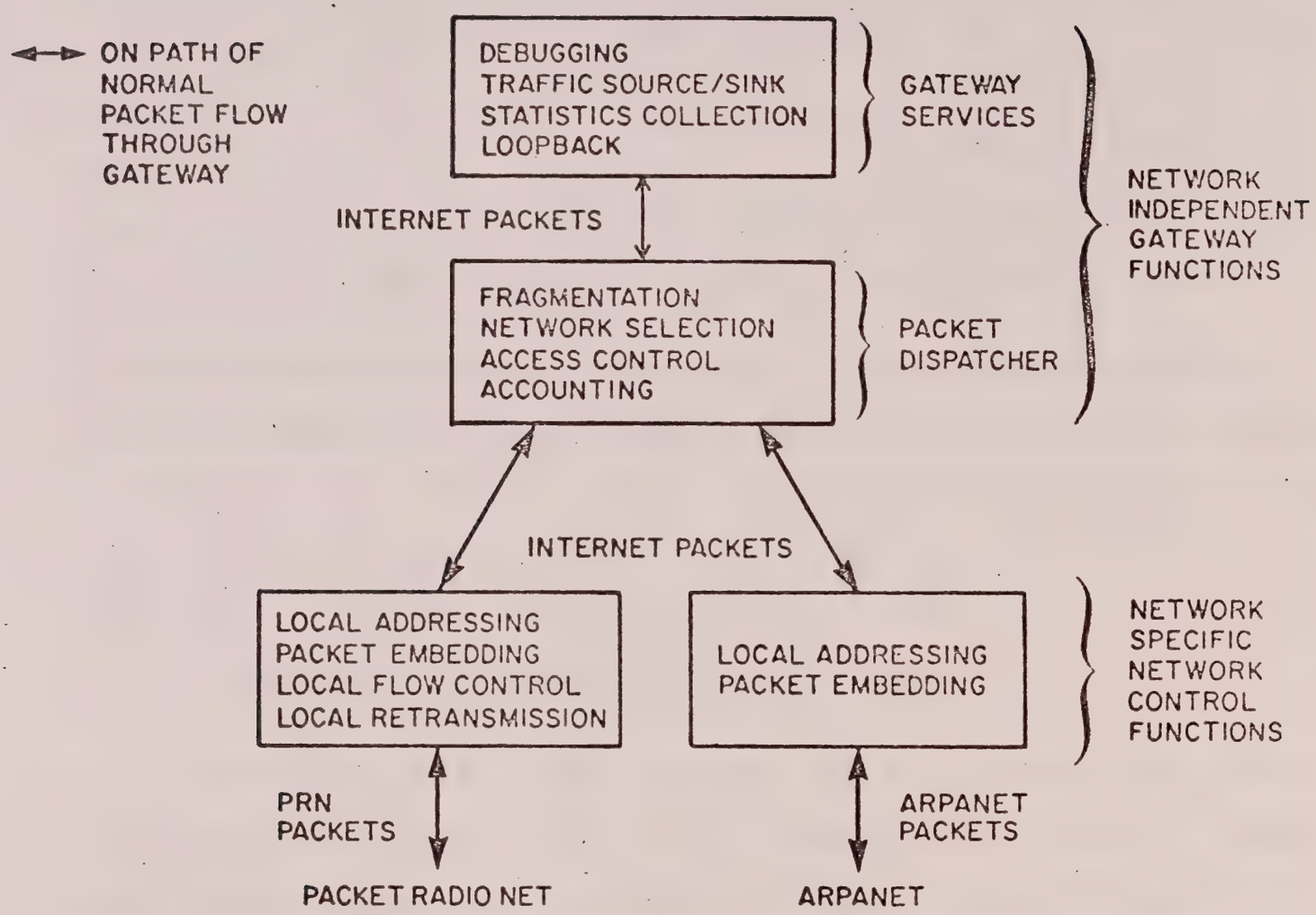


Figure 5 -- Organization of ARPANET-PRN Gateway

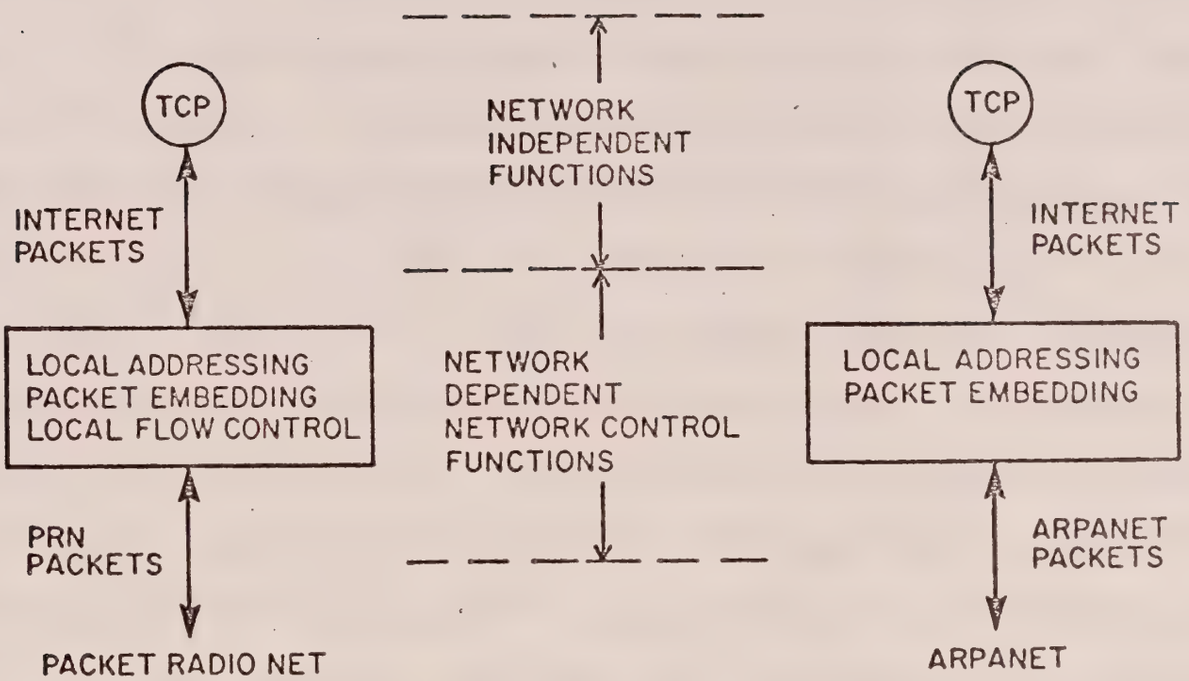


Figure 6a -- PRN Host Figure 6b -- ARPANET Host

7. Debugging and Coordination

The code in gateways will require debugging from time to time. Even in the absence of problems with gateway routines, operating parameters will require modification as new gateways are added or characteristics of constituent networks change. The gateway monitoring and coordination center is an appropriate location for performing these tasks. The vehicle for sending debugging messages between gateways and the GCC is raw internet packets. The debugging routines in each gateway appear as a fake host able to send and receive raw internetwork packets. These packets are analogous to the raw network messages of the ARPANET; as little processing as possible is performed as these packets are propagated through the gateway network. The only field of the internetwork header which absolutely must have meaning for purposes of internetwork routing is the destination identifier. In a debugging message the contents of this field specify the gateway which is being debugged, reloaded, or whatever. Responses to debugging packets will specify the GCC as destination. The text of the packet identifies the action to be performed, such as examining an area of memory. In this case the text of the response contains the contents of that area. While the debugger fake host need not have a full TCP (that is, it will not be required to interpret the majority of the header fields),

the debugger itself can take advantage of the existence of these fields. The source identifier field provides a convenient "return address" with which the debugger in any gateway can send its response to the GCC, and the GCC can identify the gateway which is responding. The sequence field provides detection and elimination of duplicate packets. This prevents a deposit command, for example, from being executed twice if some network along the way duplicates the original packet.

So that the debugger in each gateway need not have a TCP associated with it, the GCC debugging routines are restricted to permit at most one outstanding (unacknowledged) message to any given gateway. In addition debugging packets must not be fragmented since without a TCP the debugger in the destination gateway cannot reassemble the original packet. This is achieved through use of the INWG experimental standard [Cerf 74c] which states that constituent networks should accept internetwork packets up through 255 octets in length not including local network headers. Raw internet packets will also be used for sending statistics gathered at each gateway to the GCC.

Three additional functions which may reside in any gateway are data source, data sink, and loop-back capabilities. Although not precisely part of debugging or statistics functions, these are related to both. They provide test facilities to monitor

performance of the gateway network. Faults may be traced by "bouncing" packets off gateway loop-backs successively more distant from the GCC.

We have specified that raw internet packets contain a destination address in the same fashion as normal internet packets. We have suggested that the text field and some internet header fields may be used by the debugger routines in gateways and in the GCC. Beyond this we leave the format of debugging packets intentionally unspecified. The format and the debugging protocol are private to the gateway debugging function and need not be consistent with normal data transmission protocol except to permit delivery of the packet.

8. Backwards Compatibility

Although gateways permit the interconnection of networks and communication across network boundaries, the ability of a host on a foreign network to use the facilities of a local network is limited by the form of network interconnection used. Since all communication across network boundaries, as described in previous sections of this report, must be in the form of TCP segments, it is impossible for a foreign host to participate in the same levels of communication in which a local host might participate. There are some instances where it is desirable to permit a foreign host to communicate at (or near) the host/network protocol level. For example, a local host may communicate with other, less powerful hosts in the lowest level of host/network protocol, but a foreign host cannot communicate with those hosts no matter how hard it tries or no matter how willing it is to accept inefficiencies in the communications. In this case, the less powerful host may be provided by the network itself (such as the statistics hosts in an ARPA Network IMP [BBN 74]), and it may be impossible to have it understand the internetwork protocol.

Remote use of the host/network protocol can be made possible by installing a TCP service on each network. To send a message in the format of the destination network, the foreign host would embed the destination network message in an internetwork segment

addressed to the TCP service on the destination network. At that TCP, the internetwork header and trailer would be removed, leaving a message in the proper format for the destination network. The TCP would then insert any header information necessary to identify the foreign source of the message so that replies to the message may be embedded in internetwork framing and returned to the foreign host.

To communicate with some host on the local network, the foreign host should set up a "mailbox" at the local TCP service. The foreign host would then notify the local host of the address of the mailbox to be used. Messages originating in the local host would then be sent to the mailbox; the TCP service would forward them to the foreign host.

This type of connection is much less efficient than the methods described in the previous sections; however, it provides a capability which is not available in any other way. While the TCP service could reside in the gateway machine, and even share code with gateway routines, the two functions are logically distinct and may be physically so as well.

Appendix A -- Relationship Between Host Routing and Internetwork Routing

The ARPA Network currently uses a routing algorithm wherein packets are fundamentally routed to IMPs and then the IMPs sub-route the packets to the actual host on the IMP as specified in the network address carried in the packet. It has been suggested that by switching to a system of host routing, internetwork routing would be significantly facilitated. By "host routing" is meant a system wherein the basis of the network routing calculation is the "distance" to each host rather than only to each IMP as is presently done in the ARPA Network.

We have argued above that gateway supernetwork routing is necessary for internetworking as it will be impossible for all networks to be made to use identical routing algorithms and to make it work. In spite of this, host routing is useful for two reasons. First, host routing permits use of dual homing. This would permit a host connection to a network to be more reliable since the host could be connected to more than one network node. More specifically, the connection of a gateway (itself a host) to a network could be made more reliable. Second, host routing permits more than one gateway (host) to appear to the nodes within a network as identical paths to another network. That is,

without requiring a given host within a network to know the addresses of two different gateways to another network, the given host could, in fact, take advantage of the availability of two different gateway paths to the other network. Of course, this latter use is less of an advantage when the destination network is more than one network away from the source network. In the specific case of the ARPA Network, we have considered how host routing might be provided so its advantages are available without undue implementation cost.

It is unlikely that a majority of hosts on the ARPA Network would want to take advantage of host routing in the near future. Thus, a method of providing host routing is desirable which only increases routing cost on an incremental basis as additional hosts use host routing rather than from the outset incurring the large cost in bandwidth and space used for routing by making host routing available to every host.

A likely method of making host routing available on such an incremental cost basis is to divide the routing computation into two parts, one part for the majority of hosts using IMP routing and one part for the hosts desiring host routing. To explain how this is possible, recall how the IMP routing calculation is presently done. Each IMP periodically propagates its estimate of the path of least distance to each other IMP, marking the

distance to itself as being of zero length. If in addition to zeroing its own entry in the propagated routing table the IMP also zeroed another entry (e.g., the entry of a non-existent IMP), the network routing algorithm would be able to route traffic to the particular IMP by either the actual or non-existent IMP number. If this IMP were further aware that traffic arriving at it addressed to the non-existent IMP was to be specially treated, we have the basis for a form of host routing. The trick is to give the few hosts which desire host routing (virtual) IMP numbers in addition to their actual IMP number and host address implicit in the normal network routing algorithm. Therefore, suppose that a certain host desires to be dual homeable on either of IMP A or B. Then, both IMPs A and B, in addition to zeroing entry A and B (respectively) in the routing table, would also (both) zero entry C, the entry for an otherwise non-existent IMP. If hosts desiring to send traffic to the dual homed host send that traffic to IMP C rather than to some host on IMP A or B, the traffic will take the shortest path to one of IMP A or B (each of which looks also to be IMP C). If whichever IMP actually receives the traffic, knows to send traffic arriving for IMP C to the dual homed host, we have illustrated our solution. Notice that use of this capability within the ARPA Network requires that raw (type-3) messages be used since the IMP reassembly function provided for normal messages would no longer be available.

Gateways could use this dual homing capability in several ways. A given gateway could be connected to multiple IMPs. Two different gateways connected to different IMPs but both providing a path between two networks could appear to have the same host address for purposes of routing between the two networks. Two different gateways connected to the same IMP could appear to have the same host address for purposes of internetwork routing. A little thought will show that there are cases when each of these uses is convenient and possibly saves implementation effort in other areas.

Appendix B -- Fragmentation

If an arriving packet is larger than the specified maximum packet size in the destination network, the packet must be fragmented before being forwarded to the destination network. In fragmenting the packet, the internetwork header of the original packet will have to be modified and an internetwork header will have to be set up for all fragments of the original packet. The internetwork header on all packets formed by fragmenting the incoming packet will be the same as the original header with the following exceptions:

1) The length of text field will be modified to indicate the length of text in octets in the fragment.

2) The sequence number on the first packet formed from the incoming packet will be the same as that of the incoming packet. Sequence numbers of subsequent packets formed from the original will be equal to the original sequence number plus the sum of the length of text fields of all previous fragments. An exception to this is the case in which the SYN or INT bits (see below) or both are on in the incoming packet. In this case, the sequence number of the packet fragment after the first packet in the segment must be increased by one.

3) Some of the control fields in the internet header are:

SYN: Request to synchronize sending sequence numbers

ACK: There is a valid Acknowledgement in the 32 bit ACK field

FIN: Sender will stop SENDing and RECEIVEing on this connection

DSN: The sender has stopped using sequence numbers and wants to initiate a new sequence number for sending

EOS: This packet is the end of a segment and therefore has a checksum in the 16 bit checksum field. If this bit is not set, the 16 bit checksum field is to be ignored. The bit is usually set, but if segmentation at a GATEWAY occurs, the packets preceding the last one will not have checksums, and the last packet will have the checksum for the entire original segment as it was calculated by the sending TCP.

EOL: This packet contains the last segment of a letter. The EOS bit will always be set in this case.

INT: The sender wants to INTERRUPT on this connection.

CD: Control dispatch field specifies various TCP control functions such as null, event, special (reset, echo, query, status, discard, etc.), reject, etc.

The control information will be modified as follows:

If the SYN bit is on in the original packet, it will be on in the first packet formed and off in all subsequent packets.

If the ACK bit is on in the original packet, it will be on in all packets formed. Note that this implies that each packet formed will have a valid acknowledgement in the ack field. This acknowledgement will be the acknowledgement in the original packet.

If the FIN bit is on in the original packet, it will be on in the final packet formed and off in all other packets.

If the DSN bit is on in the original packet, it will be on in the final packet formed and off in all other packets.

If the EOS bit is on in the original packet, it will be on in the final packet and off in all other packets. Thus the value in the checksum field will only be considered as a valid checksum in the final packet formed.

If the EOL bit is on in the original packet, it will be on in the final packet formed and off in all other packets.

If the INT bit is on in the original packet, it will be on in the first packet formed and off in all subsequent packets.

If the CD field indicates special function (010) in the original packet, it will be the same in the first packet formed and null (000) in all subsequent packets. Thus, the value in the control data octet will only be interpreted in the first packet formed. If the CD field contains the reject (011) or event (001) codes, the packet should not contain any text, therefore it will be unnecessary to fragment it.

Note that by setting the control bits as above, the internetwork header may be copied from the original packet to all fragments with the exception of the length of text, sequence number and control fields. In particular the ack, control data and checksum fields need not be modified from the original internetwork header as their interpretation will be governed by the values of the control bits as outlined above.

Appendix C -- An example of network specific functions

This section describes an example of the network specific network control functions which are planned for a particular gateway implementation. The gateway is between the packet radio network (PRN) and the ARPANET, and the functions described are on the PRN portion of the gateway.

The gateway program communicates with the ARPANET via the experimental network control program (XNCP) and with the PRN via the connection program. To communicate with the ARPANET, the gateway program uses the "QSPEC" ELF operating system XNCP call. This call takes as parameters a two word mask and two word header as specified by the Host-to-Host protocol and returns send and receive interprocess communication port (IPP) numbers. As messages are received by the XNCP from the gateway process IPP or the IMP input device, they are masked and compared to the header value specified in the QSPEC call. All messages matching the header specified are passed to the IMP output device or to the gateway program IPP. Thus, on the ARPANET side, by specifying a mask and header to allow regular messages from any Host on any link, the gateway program can receive and send all messages via one connection consisting of an input-output IPP pair.

To communicate with the PRN, the gateway program uses the Connection-Open emulator trap (EMT) call. The receive IPP number, destination device serial number, source and destination process identifiers (PIDs), connection type (sequenced or non-sequenced) and number of buffers desired are specified in this call and a send IPP number is returned. A separate connection consisting of an input-output IPP pair is maintained for each gateway to PRN device and process pair. Connections are opened in two cases: 1) a message is received from the ARPANET destined for a packet radio device, 2) a message is received from a packet radio device destined for the ARPANET. In the first case, the PRN device serial number and destination process id may be obtained from information in the internetwork packet header. These values can then be used in a Connection-Open EMT call. In the second case, a wild card Connection-Open call will be maintained to receive messages from the PRN. Once a message has been received from a PRN device, a Connection-Open call can be issued with parameters specifying that device. In either case, a non-sequenced connection is requested as the gateway-to-gateway protocol does not require sequencing of packets on a connection.

The number of buffers requested in the Connection-Open call is the number of buffers to be allocated by the connection

program for each of the receive and send IPPs. The buffering allocation algorithm may in fact assign fewer buffers or decrease the number of buffers assigned dynamically according to the availability of buffer space and window size. Thus the number of buffers desired is an upper bound on the number of buffers actually assigned. Each buffer will consist of one status word used by the dispatcher and 140 words of packet formatted according to the standard data transmission protocol. When the dispatcher receives an internet packet, it will request an internet buffer. If a buffer is available, the packet will be passed to the gateway program via an IPP. If an internet buffer is not available, the dispatcher will discard the packet. When it receives a packet, the gateway will re-address it and if necessary open a new connection as outlined above. The gateway will then transmit the buffer either to the ARPANET via the XNCP or to the PRN via the connection program. By appropriate manipulation of buffer pointers it should be unnecessary for the gateway program to copy the packet text between buffers except when fragmentation is necessary.

We wish PRN operation to withstand packet error rates as high as ten percent. To avoid congestion in the PRN, intermediate acknowledgement of internetwork traffic within the PRN is thought necessary. For this reason, a header field value

has been defined for internet acknowledgement (use = gateway-gateway, ack = 1).

Measurements of traffic through the gateway will be kept in order to analyze gateway performance. For traffic to and from the PRN, the connection program will maintain the following statistics:

count of received packets ahead of window (where the window is the bit mask used by the connection program for filtering duplicates)
count of received packets in window
count of duplicates
count of packets flushed for lack of buffers
count of packets transmitted
count of re-transmissions due to lack of end-to-end ack
minimum time to get end-to-end ack
maximum time to get end-to-end ack
end-to-end ack wait for last ack received

For traffic to and from the ARPANET, the gateway program will maintain the following statistics:

count of packets transmitted
count of packets received
count of packets flushed for lack of buffers

Several extensions of the gateway are planned in order to provide gateway to gateway functions such as dynamic routing, access control (which Hosts on one network may access which Hosts on another network), and accounting for network use.

References

BBN 74.
Bolt Beranek and Newman Inc., "Specifications for the Interconnection of a Host and an IMP," BBN Report 1822, revised December 1974.

BCPL 74.
"The BCPL Reference Manual," Bolt Beranek and Newman, September 1974.

Belloni 74.
A. Belloni, M. Bozzetti, and G. Le Moli, "Routing and Internetworking," International Working Group Protocol Note 10, August 1974 and February 1975.

Belsnes 74.
D. Belsnes, "Flow Control in Packet Switching Networks," INWG Note No. 63, October 1974.

Burchfiel 74.
J. Burchfiel and R. Tomlinson, "An Experimental Simulation of a Satellite Gateway," INWG Experiment Note 2, August 1974.

Cerf 74a.
V.G. Cerf and R.E. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Transactions on Communications, Vol. COM-22 5, May 1974, pp. 637-648.

Cerf 74b.
V. Cerf, Y. Dalal, and C. Sunshine, "Specification of Internet Transmission Control Program," INWG Note 72, revised December 1974.

Cerf 74c.
V. Cerf, "Minutes of Stockholm INWG Meeting," INWG Note 73, December 20, 1974.

Crowther 72.
W.R. Crowther and D.C. Walden, "Response to INWG Note 6," INWG Note 10, December 1972.

Crowther 75.
W.R. Crowther, F.E. Heart, A.A. McKenzie, J.M. McQuillan, and D.C. Walden, "Issues in Packet-Switching Network Design," to be presented at the AFIPS 1975 National Computer Conference.

Kuo 75.
F.F. Kuo, "Political and Economic Issues for Internetwork Connections," Computer Communication Review, Vol. 5, No. 1, January 1975, pp. 32-34.

Lloyd 75.
D. Lloyd, M. Galland, and P. Kirstein, "Aim and Objectives of Internetwork Experiments," INWG Experiment Note 3, January 1975.

McKenzie 72.
A.A. McKenzie, B.P. Cosell, J.M. McQuillan, and M.J. Thrope, "The Network Control Center for the ARPA Network," Proceedings of the First International Conference on Computer Communications, pp. 185-191, October 1972.

McKenzie 74a.
A.A. McKenzie, "Some Computer Network Interconnection Issues," Proceedings of the AFIPS 1974 National Computer Conference, pp. 857-859, May 1974.

McKenzie 74b.
A.A. McKenzie, "Internetwork Host-to-Host Protocol," INWG Note 74, December 1974.

McQuillan 74.
J.M. McQuillan, "Adaptive Routing Algorithms for Distributed Computer Networks," BBN Report 2831, May 1974.

Mader 74.
E. Mader, W. Plummer, and R. Tomlinson, "A Protocol Experiment," INWG Experiment Note 1, August 1974.

NAC 73.
Network Analysis Corporation, "Comparison of Hop-by-Hop and End-to-End Acknowledgement Schemes," Packet Radio Temporary Note 7, January 1973.

Opderbeck 74.
H. Opderbeck and L. Kleinrock, "The Influence of Control Procedures on the Performance of Packet-Switching Networks," National Telecommunications Conference Proceedings, San Diego, December 1974.

Zimmerman 74.
H. Zimmerman and M. Elie, "Transport Protocol. Standard Host-Host Protocol for Heterogeneous Computer Networks," INWG Note 61, April 1974.